

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-131168

(43)Date of publication of application : 13.05.1994

(51)Int.Cl. G06F 9/06

G06F 12/00

G06F 15/20

G06F 15/60

(21)Application number : 04-306223

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 20.10.1992

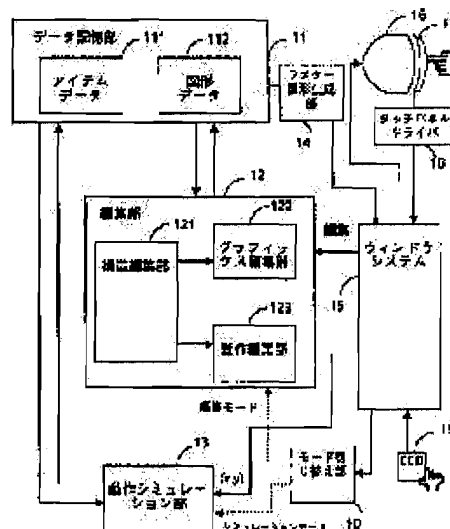
(72)Inventor : KOSHIRO HIROYUKI

(54) USER INTERFACE PREPARATION SUPPORTING DEVICE

(57)Abstract:

PURPOSE: To easily perform the graphical design of the user interface(UI) part of an equipment and the design of operations and to easily perform the operation simulation of the prepared design of the UI user interface part.

CONSTITUTION: This UI preparation supporting device is for designing the UI part of the equipment by parts provided with predetermined attribute information and is provided with a graphics editing part 122 for editing the graphic information of the parts and generating the graphic information by which the UI part is plotted and an operation editing part 123 for generating description for instructing the operations for the respective parts provided with the generated graphic information. Also, the operation simulation can be performed for a designed result.



[0041] The action script interpreting part 133 that is the interpretation module of a simulation interprets the descriptor (command) of an action script. In accordance with the result of the interpretation, the action script interpreting part 133 gives an instruction to the item managing part 132, and sets and refers to the timer table 134, the variable table 135, and the like. Items whose action can be described with an action script are respective items such as a button, a selector, and a cascade, each having a trigger region, and a timer item that operates depending on time. Fig. 12 shows an example of descriptors (commands) prepared in the action script. This example includes 21 types of descriptor. Some commands have an argument and the others do not. In either case, an interpretation unit (sentence) is made by putting a command between two marks "(" and ")". A plurality of sentences can be arranged in an item, and they are interpreted in order starting from a left side. A description is input within an item property window that is opened while being edited. The consistency of the marks "(" and ")" is checked when an input is settled. The action script interpreting part 133 invalidates a sentence having an improper argument and/or an uninterpretable command. The action script interpreting part 133 also displays an error on a console and proceeds to the interpretation of the following sentence. All variables in the script are held as character strings (arbitrary length), and they are converted to numeric values as necessary. All character strings that cannot be converted are converted to "0". According to this method, an unpredictable evaluation and the stop of interpretation, which are caused by a description error that does not make sense, can be reduced.

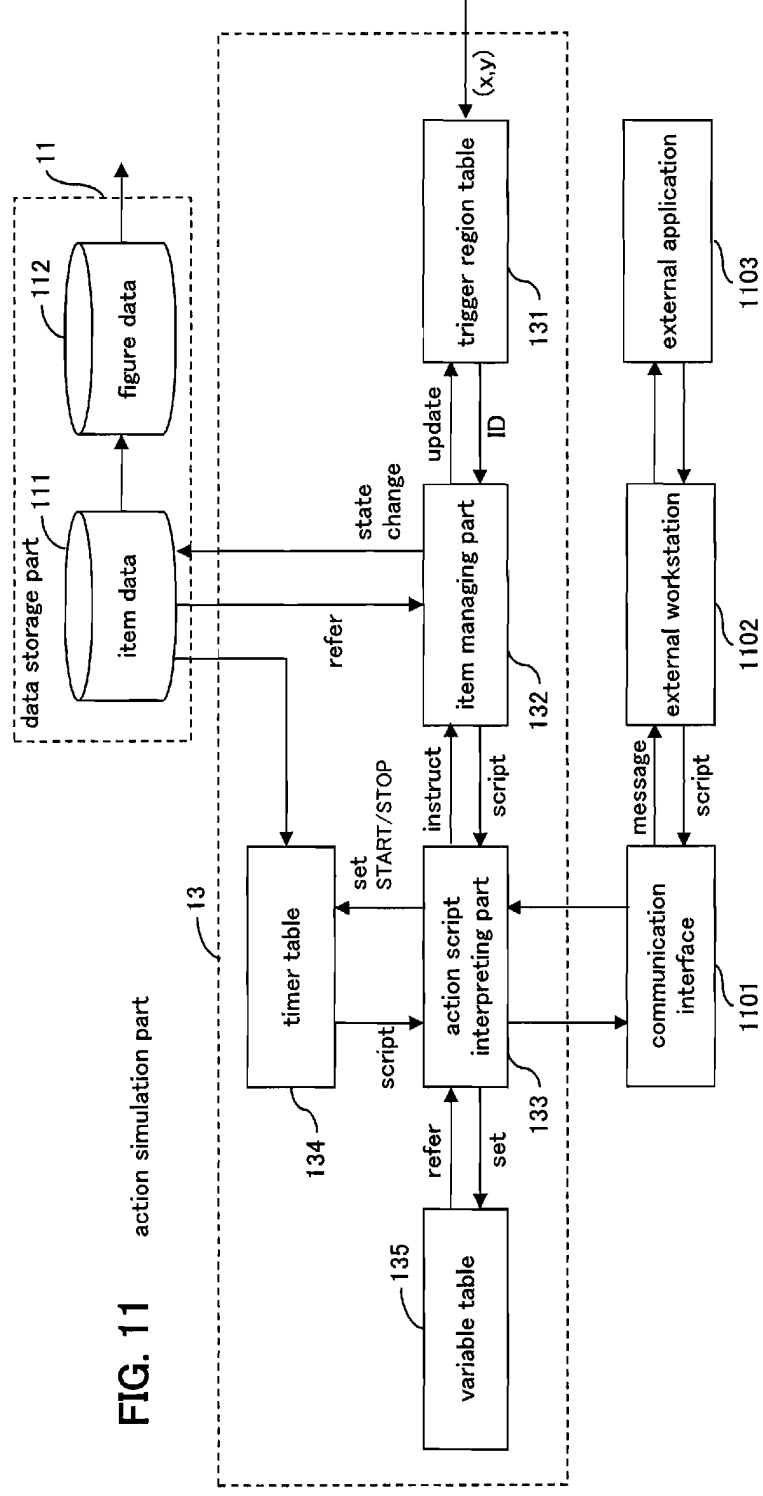


FIG. 11 action simulation part

FIG. 12 List of the descriptors (commands) of an action script

classification	descriptor	outline
frame change	Jump	Move current FRAME
	Popup	Move to a child FRAME
	Close	Move to a parent FRAME
switch figure pointer	Chg	Change the pointer of a figure
	On	Macro for setting the pointer of a figure to "1"
	Off	Macro for setting the pointer of a figure to "0"
	Default	Set the pointer of a figure to be an initial value
variable operation	Set	Set a value for a variable
	Shift	Shift a variable up by one decimal digit
	Save	Change a subtrahend from a local to a global
	Cancel	Cancel the variable of a local scope
indirect operation	Do	Evaluate the specified script
syntax	If...else...	Conditional branching of interpretation
	Return	Suspension of interpretation
input/output	Open	Specify a file
	Input	Input from a file
	Output	Output to a file
	Read	Input from local memory
	Write	Output to local memory
event	Flush	Forced output of display
	Retouch	Re-storing of an event

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-131168

(43)公開日 平成6年(1994)5月13日

(51)Int.Cl. ⁵	識別記号	片内整理番号	F I	技術表示箇所
G 0 6 F 9/06	4 3 0 P	9367-5B		
		E 9367-5B		
12/00	5 1 5 B	8526-5B		
15/20		D 7052-5L		
15/60	3 6 0 P	7922-5L		

審査請求 未請求 請求項の数6(全 18 頁)

(21)出願番号 特願平4-306223

(22)出願日 平成4年(1992)10月20日

(71)出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂三丁目3番5号

(72)発明者 小城 浩之

神奈川県横浜市保土ヶ谷区神戸町134番地

横浜ビジネスパークイーストタワー 富

士ゼロックス株式会社内

(74)代理人 弁理士 岩上 昇一 (外3名)

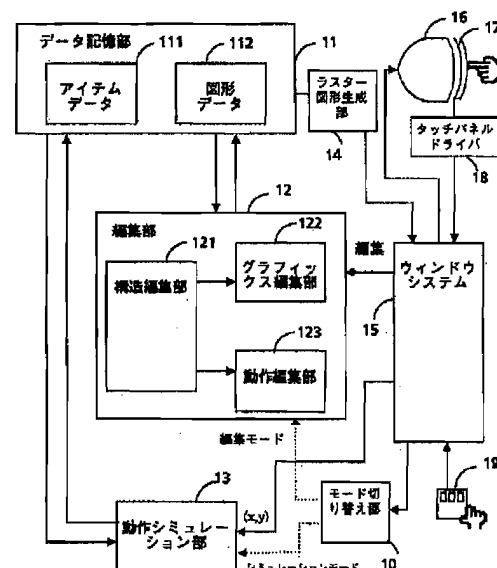
(54)【発明の名称】 ユーザインタフェース作成支援装置

(57)【要約】

【目的】 機器のユーザインタフェース (UI) 部のグラフィカルなデザインと動作のデザインを容易に行うこと。作成されたUIユーザインタフェース部のデザインの動作シミュレーションを容易に行うこと。

【構成】 機器のUI部を予め定められた属性情報を有する部品により設計するUI作成支援装置であって、部品の図形情報を編集して、前記UI部を描いた図形情報を生成するグラフィックス編集部122と、前記生成された図形情報を持つ各部品についてその動作を指示する記述を生成する動作編集部123とを備える。また、設計された結果について動作シミュレーションを行うことができる。

【図1】



【特許請求の範囲】

【請求項1】 機器のユーザインタフェース部を予め定められた属性情報を有する部品により設計するユーザインタフェース作成支援装置であって、部品の図形情報を編集して、前記ユーザインタフェース部を描いた図形情報を生成する第1の手段と、前記部品についてその動作を指示する記述を生成する第2の手段とを備えたユーザインタフェース作成支援装置。

【請求項2】 前記部品として、①ボタンなど操作の対象になる「触る部品」と、②メッセージ、状態などの変化を表す「表示する部品」と、③上記①、②の部品を配置する背景となる「背景になる部品」とが予め用意されていることを特徴とする請求項1記載のユーザインタフェース作成支援装置。

【請求項3】 前記予め用意されている部品は、次の6種類であることを特徴とする請求項2記載のユーザインタフェース作成支援装置。

- ① 他の部品の背景となる部品
- ② フレーム上に動的な変数の表示を行う部品
- ③ 1つのトリガー領域を設定でき、トリガー領域の操作によりオンとオフの状態を切り替える部品
- ④ 1つのトリガー領域を設定でき、トリガー領域の操作により排他的に一つを選択可能なN個の状態を持つ部品
- ⑤ 2つのトリガー領域を設定でき、これらのトリガー領域の操作により排他的に一つを選択可能なN個の状態を持つ部品
- ⑥ 定期的に動作を実行可能にする部品

【請求項4】 設計に用いられた部品を樹木構造に関係づけて管理する第3の手段を設けたことを特徴とする請求項1記載のユーザインタフェース作成支援装置。

【請求項5】 前記各部品は、固有な属性情報を有し、その属性情報は図形情報と機能情報と構造情報を含むことを特徴とする請求項1記載のユーザインタフェース作成支援装置。

【請求項6】 請求項1記載のユーザインタフェース作成支援装置において、第1、第2および第3の手段により設計された機器のユーザインタフェース部の動作のシミュレーションを行うために、第2の手段により生成された前記動作を指示する記述に基づいて、第1の手段により生成された図形情報の表示を制御する第4の手段を設けたことを特徴とするもの。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、例えば複写機、ファクシミリ装置、卓上計算機、電話機、テレビ、ビデオデッキ、カーステレオ、洗濯機などの種々の分野の電氣的製品における操作部（コントロールパネル）のデザインと

動作を設計し、動作シミュレーションを行うユーザインタフェース作成支援装置に関し、製品のユーザインタフェースコンセプトのプレゼンテーション、ユーザインタフェース画面の操作性の模擬テスト、機器の制御、情報の検索などの広範囲の用途に利用できるものである。

【0002】

【従来の技術】複写機における機能の拡大、デジタル化、カラー化、FAXとの複合化等は、単にコピーをとるという枠を越えて、複写機の目の前に立って原稿を加工編集する創作的な空間をも提供しつつある。しかし、一方では、ユーザは従来通りの操作と概念のもとで早く且つミスすることなしに“コピー”という作業をすめたいと期待している。予測のつかない操作手順はユーザに威圧感を与え、複写機の機能とは裏腹に、その評価は決して良いものとは言えなくなってしまう。機能の充実、オフィス空間を演出する高品位な外観デザインに加えて、コントロールパネルといわれる操作部のデザインと操作性品質の向上を総合的に検討していかなければ、複写機の真価をユーザに提供することはできない。このような問題は、複写機のみに限らず、ユーザが操作するユーザインタフェース部を有するその他の事務用機器やあらゆる電化製品にも共通するところである。そこで、近年、コントロールパネル上での増大する操作ボタンの数とそのレイアウトの問題を解決する方法として、液晶パネルとタッチパネルを組み合わせた対話型ユーザインタフェース（UI）が用いられるようになってきた。

【0003】一方、アプリケーションプログラムのグラフィックユーザインタフェースを作成するための作成支援ツールは、従来から知られている（例えば、特開平2-254539号公報）。

【0004】

【発明が解決しようとする課題】前記液晶パネルとタッチパネルの対話型ユーザインタフェースの設計に際しては、操作に応じて表示の変化するグラフィックスとタイミングを検討するために、デザイナーは各変化の時点ごとのデザインを紙上に多数描かなければならず、この作業は時間のかかる煩雑なものであった。また、そのような多数の紙上に描いた静止した画像からデザインされた製品の動作の推移を把握し、検討するのも容易ではなかった。また、前記アプリケーションプログラムのグラフィックユーザインタフェースを作成するための作成支援ツールは、プログラムの作成を目的するものであり、機器のコントロールパネルのようなユーザインタフェースの作成、検討のための支援装置として適用できるものではなかった。

【0005】本発明は、このような問題点を解決することを課題とするものである。即ち、本発明は、機器のユーザインタフェース部のグラフィカルなデザインと動作のデザインを容易に行うことのできるユーザインタフェース作成支援装置を提供することを目的とするもので

ある。また、本発明は、上記作成されたユーザインタフェース部のデザインの動作シミュレーションを容易に行うことのできるユーザインタフェース作成支援装置を提供することを目的とするものである。

【0006】

【課題を解決するための手段】本発明は、機器のユーザインタフェース部を予め定められた属性情報を有する部品により設計するユーザインタフェース作成支援装置であって、部品の図形情報を編集して、前記ユーザインタフェース部を描いた図形情報を生成する第1の手段(図1の122)と、前記各部品についてその動作を指示する記述を生成する第2の手段(図1の123)とを備える。

【0007】本発明の一態様においては、前記部品として、①ボタンなど操作の対象になる「触る部品」と、②メッセージ、状態などの変化を表す「表示する部品」と、③上記①、②の部品を配置する背景となる「背景になる部品」とが予め用意される。

【0008】本発明の他の態様においては、前記予め用意されている部品として、次の6種類とすることのできる。

- ① 他の部品の背景となる部品(フレームアイテムFRAME ITEM)
- ② フレーム上に動的な変数の表示(例、メッセージなどの表示)を行う部品(インディケータアイテムINDICATOR ITEM)
- ③ 1つのトリガー領域を設定でき、トリガー領域の操作によりオンONとオフOFFの状態を切り替える部品(ボタンアイテムBUTTON ITEM)
- ④ 1つのトリガー領域を設定でき、トリガー領域の操作により排他的に一つを選択可能なN個の状態を持つ部品(セレクトアイテムSELECTOR ITEM)
- ⑤ 2つのトリガー領域を設定でき、これらのトリガー領域の操作により排他的に一つを選択可能なN個の状態を持つ部品(カスケードアイテムCASCADE ITEM)
- ⑥ 定期的な動作を実行可能にする部品(タイマアイテムTIMER ITEM)

また、本発明において、前記部品は、属性情報として図形情報、機能情報、構造情報を含む。具代的には例えば図4に示すように、図形データ、トリガー領域、アクションスクリプト、下層アイテムへのポインタ…などである。

【0009】また、本発明は、上記第1および第2の手段を有する構成において、設計に用いられた部品を樹木構造に関係づけて管理する第3の手段(図1の121)を設けることができる。

【0010】また、本発明は、前記第1、第2および第3の手段により設計された機器のユーザインタフェース部の動作のシミュレーションを行うために、第2の手段

により生成された前記動作を指示する記述に基づいて、第1の手段により生成された図形情報の表示を制御する第4の手段(図1の13、図11の13)が設けられる。

【0011】

【作用】本発明は固有の属性を有する部品が用意され、この部品を基に機器のコントロールパネルのようなユーザインタフェース部を設計する。その設計において、デザイナーは、第1の手段(グラフィックス編集部122)を用いて、背景となる部品の図形の上に他の所望の属性項目を持つ部品の図形を選択配置し、描画編集することによりユーザインタフェース部の所望の外観の画像データを生成する。また、第2の手段(動作編集部123)を用いて、部品の属性項目を設定するとともに、必要な動作の記述(アクションスクリプト)を作成することにより動作の設計を行う。本発明によれば、描画情報だけでなく動作情報をも設定可能な部品を用いて、動作をも含めたユーザインタフェース部の描画編集を行うことができ、操作に応じて状態の変化する動きのあるユーザインタフェース部の設計を容易に行うことができる。

【0012】部品としては、基本的には背景になる部品、触る部品、表示する部品があり、実際の設計を行うには前述のわずか6種類の部品を用意するだけで、広範な分野で状態の変化を伴うユーザインタフェース部の設計が可能である。

【0013】また、第3の手段(構造編集部121)により部品は樹木構造に関係づけられる。樹木構造に関係づけられることにより、複雑な機能のユーザインタフェースの設計も可能となる。

【0014】本発明において、第4の手段(動作シミュレーション部13)は、前述のように部品を用いた設計により得られたユーザインタフェース部の部品データにより、動作シミュレーションを行うことができる。座標入力により部品が指示されると、その部品に関する動作を指示する記述(アクションスクリプト)を解釈し、実行することにより画面上で動きをシミュレーションすることができる。即ち、デザイナーは、製品コンセプトに従ってインタフェースの部のデザインを作成し、その作成の途中でも、あるいは作成終了後直ちに第4の手段によりシミュレーションを行うことが可能である。従って、動作をも含めたデザインの善し悪しをすぐに評価し、デザインの修正を行うという検討作業が可能となり、品質の高いユーザインタフェース部を時間をかけずに設計することができる。

【0015】更に、第4の手段(動作シミュレーション部13)を有することにより、本発明はユーザインタフェース部の設計結果を、関係者に対し分かりやすいプレゼンテーションを行うことができる。従来はこのようなプレゼンテーションでは、操作部の動きの状態ごとの多数の図面を作成し、時間順序に従って逐一説明する必要

があったが、ユーザインタフェース部の設計結果が実際の製品と同じ外観が画面表示され、疑似的に操作してみることができ、その動作をも見ることができ、効果的なプレゼンテーションを行うことができる。

【0016】

【実施例】図1は本発明の一実施例の構成の概略を示す図である。この実施例は、図1に示すように、予め用意された操作部の部品のデータであるアイテムデータ111やそのアイテムデータを基に作成、編集された操作部の図形データ112などの情報を記憶するデータ記憶部111と、構造データ、グラフィックスおよび動作記述（アクションスクリプト）の作成、編集を行う編集部112と、作成された操作部のデザイン情報による動作のシミュレーションを行う動作シミュレーション部113と、編集あるいは動作シミュレーションのために図形データをラスター図形に変換するラスター図形生成部114と、ウィンドウシステム115と、図形データを表示する表示部116と、編集操作の入力や指示位置情報を入力するための入力手段であるタッチパネル117およびそのドライバ118と、マウス119を備えている。編集部112は、アイテムの構造の生成、編集を行うと共にアイテムの管理を行う構造編集部1121と、部品アイテムの図形情報を編集して、操作部の図形情報を生成するグラフィックス編集部1122と、そのグラフィックス編集部1122により作成された図形情報における各部品の動作を指示する記述（アクションスクリプト）を生成する動作編集部1123とを備えている。

【0017】この実施例では、編集部112および動作シミュレーション部113は、ウィンドウシステム115を用いたソフトウェアによって構成されるが、それを実行するためのハードウェアの構成例は、図2に示すように、編集部112によるデザイン編集時には、ワークステーションとディスプレイ装置とポインティングデバイス（マウス）からなり、動作シミュレーション部113によるシミュレーション時には操作指示部としてRS-232Cを介してタッチパネルとそのコントローラが更に実装される。

【0018】ユーザが編集操作するこの実施例の装置におけるグラフィカルユーザインタフェース（GUI）は、次の3つに大別される。

- ① メインウィンドウ（アイテムのレイアウト編集、図形の編集およびシミュレーション）
- ② ツリーウィンドウ（アイテムの構造表示、編集）
- ③ アイテムプロパティウィンドウ（アイテムの振る舞いの編集）

図14は上記3つのウィンドウの一例を示し、中央の一番大きなウィンドウがメインウィンドウ、右側の縦長のウィンドウがツリーウィンドウ、左下の小さなウィンドウがアイテムプロパティウィンドウである。アイテムおよび図形については後述するが、メインウィンドウとツ

リーウィンドウは、異なる観点から、データを表示したものであり、1対1に対応している。メインウィンドウは、編集とシミュレーションを兼ねている。ツリーウィンドウは、必要に応じて任意に表示／取消しができる。アイテムプロパティウィンドウは、アイテムの振る舞いを編集する場合に任意に表示する。編集対象のアイテムは1つに特定されるため、同時に2つ以上は表示されない。ツリーウィンドウとアイテムプロパティウィンドウを表示した状態で、シミュレーションをおこなうことが可能である。シミュレーション中のアイテムプロパティウィンドウは、動作中のアイテムに対応して同期表示をする。その他、ユーティリティウィンドウとして、図形プロパティ、カラーパレットが描画編集用に用意されている。

【0019】図形の部品化

本実施例におけるユーザインタフェース画面の部品は、いくつかの図形のまとまりで見かけ上の意味を形作っている。しかし、部品は図形情報だけでなく、機能情報を含んだ上で意味をもった部品として成立する。部品としての機能は、以下の3つに大別される。

- ① ボタンなど操作の対象になる”触る部品”
- ② メッセージ、状態などの変化を”表示する部品”
- ③ ①②の部品を配置する”背景になる部品”

この組合せによる簡単なユーザインタフェース画面を図3に示す。

【0020】部品とアイテム

本実施例では、ボタン操作式（ダイアルなどがない）のユーザインタフェース画面のデザインに注目して、6種類の部品を定義した。以後、この部品をアイテムと呼ぶ。アイテムは、名前、ID、サイズ、図形情報、現在の状態、タッチ領域（トリガー範囲）、動作情報などをもつ。以下に、6つのアイテムの詳細をしめす。

【0021】1）フレームアイテム（FRAME ITEM）

部品となるアイテムの背景になるアイテムである。子供アイテムを持てる唯一のアイテムであり、フレームアイテムを単位として、表示が制御される。フレームアイテム上に子供フレームアイテムを貼り付けてもよい。構造的には、1個の図形ポインタと子供アイテムポインタをもっている。

【0022】2）インディケータアイテム（INDICATOR）

フレーム上に動的な変数の表示をおこなうアイテムである。メッセージなどの表示に用いる。構造的には、1個の図形ポインタをもっている。

【0023】3）ボタンアイテム（BUTTON）

OFF、ONの表情をもつアイテム。トリガー領域もち、シミュレーション時には、このトリガー領域内を触れると、OFFからONまたはONからOFFの図形に表示を変更し、アイテム内に記述された動作を解釈して

実行する。構造的には、2個の図形ポイントと2個の動作情報と1個のタッチ領域をもっている。

【0024】4) セレクタアイテム (SELECTOR ITEM)

複数個 (n個) の表情をもつアイテム。状態は排他的に1つを保持し、i 番目の図形を表示する。シミュレーション時には、j 番目のトリガー領域内を触れると、i 番目から j 番目の図形に表示を変更し、j 番目に記述されている動作を解釈して実行する。編集時に個数 n を増減できる。構造的には、n 個の図形ポイントと n 個の動作情報と n 個のタッチ領域をもっている。

【0025】5) カスケードアイテム (CASCADE ITEM)

複数個 (n個) の表情をもつアイテム。セレクタアイテムと異なるのは、トリガー領域を2個もつことである。この2個のタッチ領域は、シミュレーション時に、それぞれ (i+1) 番目、(i-1) 番目に状態と表示を変更するように定義されている。また、i が1以下、n 以上になった場合に、上下限で停止するかサイクリック動作にするかの切り替えをもつ。構造的には、n 個の図形ポイントと n 個の動作情報と2個のタッチ領域をもっている。

【0026】6) タイマアイテム (TIMER ITEM)

定期的に動作を実行するアイテム。図形とトリガー領域をもたない代わりに、アイテム内に記述された動作が他のアイテムを呼び出す。構造的には、1個の動作情報と動作呼出し間隔と現在の実行中/停止中の値をもっている。

【0027】図4にそれぞれのアイテムがもつ主な情報を示し、図6にアイテムの表示例を示す。

【0028】アイテムの編集と管理

図6は編集部12の処理の流れの概略を示す図である。システムを起動し(ステップ61)、データの新規作成か既存データの編集かを決定し、既存データの編集の際には既存のデータをデータ記憶部11から読み出して編集部12へ読み込む(ステップ62)。編集部12では構造の編集(ステップ63)、グラフィックスの編集(ステップ64)、動作の編集(ステップ65)をそれぞれ行う。編集途上で、あるいは編集が終わったときに、モード切替部10によりシミュレーションモードにして動作シミュレーション部13により出来上がった部分を確認、評価することができる(ステップ67)。作成されたデータはデータ記憶部11に保存する(ステップ66)。

【0029】具体的には、ユーザは、6個のアイテムを組み合わせて、グラフィカルなデザインと動作のデザインをウィンドウを通じて対話的に進めていく。新規のデザイン作業の場合、システムは先頭になるフレームアイテムを自動的に作成する。ユーザはメインウィンドウにお

いて、この背景アイテムの枠の中に図形を描画、配置していく。この枠の中でボタンとして描かれた図形は、他の図形と区別されることはない。図形はアイテムという単位によって管理されるからである。フレーム上のボタンという機能部品は、実際に、フレームアイテムの上にボタンアイテムとして関係づけられなければならない。編集上は、フレームアイテムの上にボタンアイテムを子供として配置し、樹木構造にすることで関係付けられる。ボタンの図形は、このボタンアイテムの中に描画する。このようにして、フレームアイテムの上には複数の部品が載り、各部品は必ずいずれかのフレームアイテムを親としてもつように組み立てられていく。編集部12により構築したアイテムの樹木構造は、デザインしたユーザインタフェース画面の機能構成図を示すようになる。図7は樹木構造の一例を示すものである。この例は複写機の操作部の樹木構造を示し、図中の倍率選択画面、任意倍率、固定倍率等の名前がつけられた矩形はフレームアイテムを示している。

【0030】排他表示と画面切り替え

図7のように樹木構造で関係付けられたアイテムは、シミュレーション中の動作の局面に応じて、その内容を意図した図形に変化させなければならない。表示の変化には3通りが挙げられる。

- ① 変数により文字列表示を変更する。
- ② アイテムの図形ポイントを変更する。
- ③ アイテムを表示対象から外す。または、表示対象にする。

【0031】1) 図形ポイントと排他表示

予め描いておいた図形を切り替えるため、1つのアイテムに複数の図形ポイントをもたせておく。アイテムは、排他的に1つの図形ポイントのみを表示対象(編集対象)とするように管理をおこなう。該当するアイテムは、ボタン、セレクタ、カスケードアイテムである。ボタンアイテムの場合にはOFFの絵とONの絵が含まれ、“button(0)”と“button(1)”の表現が現在の表示対象の図形を表している。“button(0)”をOFFとするか、ONとするかの特別な区別はない。編集によって意味付けられるだけである。便宜上は0をOFFとしている。セレクタとカスケードの場合は、n個中のi番目を表示対象として、“selector(1/4)”、“cascade(3/7)”のように表現をする。

【0032】2) 画面管理と表示規則

アイテムの表示有無の機能は、フレームアイテムのみがもっている。フレームアイテムが表示対象外の場合には、フレームアイテム自身とその上に載るすべての部品アイテムが表示されない。これは、画面切り替えという動作形態を単位にしているためである。本実施例では、兄弟関係のフレームに対しては1つのみを排他的に表示する。親子関係のフレームでは、子供フレームが上に重

なる。この様子を図8に示す。表示対象は、先頭のフレームから現在操作中のフレームアイテムまでと、その中に含まれるフレーム以外の部品アイテムである。図中の矩形がフレームアイテム、円が部品アイテムを示している。また、網のアイテムが表示対象を示す。

【0033】動作の入力

複数の図形ポイントをもつアイテムは、アイテム単体での表示切り替えを内部で実現するが、アイテム間で連動した動作をおこなうには、アイテム内にその指示をもたせる必要がある。この動作指示を、アクションスクリプト（動作記述）と呼ぶ。アクションスクリプトを持つことのできるアイテムは、トリガー領域をもつアイテムとタイマアイテムである。アクションスクリプトの入力は、アイテムプロパティウインドウ（動作編集部123）中で行われる。スクリプトで指示できる動作については後述する。

【0034】組合せの自由度

デザインは、目的とするユーザインタフェース画面を作成するために、異なるアイテム構造で同じ動作結果を組み立てることができる。たとえば、図3のユーザインタフェース画面に対して、2通りの構造例を図9に示す。

(a)の場合は、[▽] [△] にボタンアイテムをそれぞれ割当てているが、(b)の場合には、カスケードアイテムの2つのトリガー領域を割当てている。それぞれに記述されるアクションスクリプトは多少異なるが、結果として、同じ動作を実現することが可能である。

【0035】描画上の特徴

液晶面を対象としたデザイン作業は、一般的なグラフィックデザインとは異なり、液晶の解像度の低さに応じた、所謂“ドット絵”の作成が主であり、“ドット絵”作成に使われるツールは、ペイントシステムが一般的である。ドットを描くデザイナーには、目的の形状を解像度によって間引き、それらしく見える“ドット絵”に形を変形して描くセンスが必要である。また、ドットに細分化された図形のプロポーショナルを変更する作業は容易ではない。本実施例では、ぬり絵としてのペイント方式ではなく、描いた図形をベクトルデータとして保持するドローイング方式を用いている。ベクトル図形とすることで、ユーザは素早い描画とレイアウト修正が可能である。本実施例が提供する描画図形を図10(a)に示す。また、ボタンなどの部品のグラフィカルなディテール（凹凸、角R）を属性（プロパティ）として与えることによって、ドットとしての書き込みを極力軽減させている。図形プロパティとして扱う要素を図10(b)に示す。図形プロパティを設定してユーザインタフェース画面をデザインすることにより、図形のディテールの表示の品位も高いものとなる。

【0036】また、実際の製品の液晶のピッチとワークステーションのCRTとのピッチが異なるために、デザインするユーザインタフェース画面に対して拡大表示が

必要とされる。拡大表示は、ベクトル座標をそのまま拡大して滑らかな図形を描いてしまったり、整数倍以外の拡大によってドットを間引いて表示したりすると、液晶格子上の“ドット絵”と異なってしまうために、デザインの検討にはならない。そこで本実施例では、ベクター図形のラスターキャンコンバージョン処理と高速な整数倍率の拡大処理を行うラスター図形生成部14を独自に備えている。Xウインドウの描画関数では、実数値計算に展開しているために品質の高いディテールを表現できないからである。

【0037】動作シミュレーション

編集モードとシミュレーションモード

モード切替え部10により編集モードとシミュレーションモードを切り替えることによって、編集モードにおいて編集部12で作成したユーザインタフェース画面データを直ちに動作させることができる。シミュレーションモードに切り替えた時点では、作成したデータの先頭アイテムからシミュレーションをおこなうが、シミュレーションにより変化したアイテムの状態は、そのままデータに反映される。したがって、その場所からの編集が可能である。

【0038】動作シミュレーション部の構成

動作シミュレーション部13は図11に示すように、トリガー領域テーブル131と、アイテム管理部132、アクションスクリプト解釈部133、タイマーテーブル134および変数テーブル135からなっている。アイテムの画面表示を切り替えるトリガーの入力手段として図1のようにマウスやタッチパネルが接続されているが、図11には図示は省略されている。なお、図11に示すようにアクションスクリプト解釈部133には、通信インタフェース1101を介して外部ワークステーション1102で外部アプリケーションプログラム1103のユーザグラフィックインタフェースのシミュレーションを行うための構成が付加されている。

【0039】トリガー領域テーブル131は、ボタン、セレクト、カスケード等のトリガー領域を有するアイテムのトリガー領域を保持し、ユーザの指定した画面上の座標がトリガー領域か否かを調べるためのものである。座標の指定は、タッチパネル17、マウス19等の入力装置からおこなわれ、座標が入力されると、その座標(x, y)がどれかのアイテムのトリガー領域に該当するかが探索され、該当するものがあればそのアイテムのIDがアイテム管理部132に送出される。表示対象であるカレントフレームのトリガー領域のみが格納され、カレントフレームが変更になると、アイテム管理部132により更新される。

【0040】アイテム管理部132は、シミュレーションを行うアイテムの管理を行うものであり、トリガーされたアイテムの情報をそのIDを基にデータ記憶部11のアイテムデータ記憶部111から読み出し、アイテム

データに含まれるアクションスクリプトをアクションスクリプト解釈部133へ与え、あるいはアクションスクリプト解釈部133からの解釈結果の指示を受けてアイテムデータの図形データのポインタの変更や文字列表示の変更、あるいはアイテムを表示対象から外しまたは表示対象にするなどのアイテムデータの状態の変更を行ったりするものである。

【0041】シミュレーションの解釈モジュールであるアクションスクリプト解釈部133は、アクションスクリプトの記述子(コマンド)を解釈し、解釈の結果によりアイテム管理部132に指示を与えたり、タイマーテーブル134や変数テーブル135の設定や参照を行うものである。アクションスクリプトで動作を記述できるアイテムは、トリガー領域を持つボタン、セレクトタ、カスケードの各アイテムと、時間で動作をするタイマーアイテムである。図12はアクションスクリプトに用意されている記述子(コマンド)の例を示すものであり、この例では記述子は21種類である。コマンドは、引数を持つものと持たないものとがある。いずれの場合も、“(”と”)”でコマンドを挟み解釈単位(文)とする。アイテム中には、文を複数羅列することができ、解釈は左から順におこなわれる。記述の入力は、編集時にアイテムプロパティウインドウを開き、その中で記述する。“(”と”)”の整合性を入力確定時におこなっている。アクションスクリプト解釈部133は、不適切な引数や解釈できないコマンドの場合には、その箇所の文を無効にする。また、アクションスクリプト解釈部133は、エラーをコンソールに表示して次の文へ解釈を進める。スクリプト中の変数はすべて文字列(任意長)として保持され、必要に応じて数値に変換される。変換できない文字列は、すべて“0”に変換される。この方法により、意味の通らない記述ミスによる予測できない評価と解釈の停止を軽減させることができる。

【0042】シミュレーション動作の流れ

以上のように構成された動作シミュレーション部の動作について説明する。シミュレーションの主な仕事は、アイテムの画面表示を切り替えることである。切り替えるトリガーとしては、マウスによる操作およびタッチパネルからの座標信号と、タイマーからの定期的な信号と、通信インタフェース1101を通じての外部からの信号の3種類がある。初期状態では、変数とタイマー動作は登録されていないので、座標信号または外部信号から動作が開始される。

【0043】1) トリガー領域と判定

受け付けたトリガー座標(x, y)が、編集により作成した各アイテム(ボタン、セレクトタ、カスケード)の持っているトリガー範囲内に入っているかどうか、“操作した”という判断の基準になる。ここでは、すべてのアイテムのトリガー範囲が対象になる訳ではない。フレームという画面単位でボタンなどの部品が配置されてお

り、画面が複数あれば、トリガー範囲も重なっている。複数のフレームには、親子関係の場合と兄弟関係の場合がある。親子のフレームは、いわゆる“ポップアップ画面”として扱われるものであり、本実施例では、子供フレームのトリガー領域のみを判断の対象としている。また、兄弟関係のフレームは、排他的に1つのみが表示されるため、判断の対象となるのは表示されているフレームのみとなる。本実施例では、最も手前(子供)に位置して且つ表示対象となっているフレームを、カレントフレームと呼ぶ。トリガーの対象となるのは、カレントフレーム上のボタン、セレクトタ、カスケードのみである。カレントフレーム上のボタン、セレクトタ、カスケードが保持しているトリガー領域は、トリガー領域テーブル131にリストアップされる。シミュレータでは、このテーブルを参照して、触った部品アイテムを割り出している。表示動作の過程でカレントフレームが変更された場合には、トリガー領域テーブル131を新しいカレントフレームのものへ更新する。

【0044】2) アクションスクリプトの解釈

トリガーの対象に該当したアイテムは、アイテム自身の状態変更と表示変更を、内部判断でおこなう。その後、アイテム中に記述されている動作(アクションスクリプト)を実行する。記述による動作は、4つに分類される。

- ① 他のアイテムの動作を指示する記述
- ② タイマー動作を扱う記述
- ③ 変数を扱う記述
- ④ 外部へ通信する記述

他のアイテムの動作を指示する記述には、直接指定アイテムの表示を切り替えるものと、指定アイテムに移行してそのアイテムのアクションスクリプトを評価するものとがある。後者は、アイテム毎に記述内容を分散して部品の振る舞いを構造化させることを可能にしている。アクションスクリプトの解釈によりカレントフレームが変更になった場合には、トリガー領域テーブル131を更新する。シミュレーション初期時には、新しい変数テーブル135とタイマーテーブル134を生成する。変数とタイマーは、それらの設定記述を含むアクションスクリプトを解釈した時点で、逐次変数テーブル135あるいはタイマーテーブル134に登録される。設定されていない変数を参照した場合には、アクションスクリプト解釈部133はその変数名を値として評価する。この評価値は、動作のデバッグに役立っている。

【0045】3) タイマー動作

タッチパネル17やマウス19からのトリガ動作以外の、定期的な動作を実現するために、タイマー機能を実装している。タイマーの用途には、点滅表示、アニメーション表示、デモンストレーションの自動実行などがある。タイマーは、設定記述を含むアクションスクリプトを解釈した時点で、タイマーテーブル134に登録され

る。一度登録したタイマーをシミュレーション中に削除することはできないが、タイマーに実行／停止を指示することができる。実行中のタイマーは、指定されたタイマーアイテムのアクションスクリプトを定期的に行行（評価）する。

【0046】4）通信動作

通信動作は、UNIXのファイルポインタ（f p）をオープンして、文字列単位で入出力をおこなう。入力用と出力用のポインタを別に割当ててあり、本実施例をパイプにしてアプリケーションと繋ぐことが可能である。出力データは、アクションスクリプトと変数によって任意の文字メッセージを組み立てて、アプリケーション側でメッセージを判断させる。入力データは、アクションスクリプトの解釈モジュールに渡される。解釈できない文字列は無効になるが、解釈可能な場合には、シミュレーション画面を外から操作することが可能である。この入出力機能によって、外部のアプリケーションのグラフィックユーザインタフェース部の働きを実現する。

【0047】以上に詳述した本実施例の装置は、機器のユーザインタフェースデザイン検討支援ツールであり、従来のアプリケーションGUI（グラフィックユーザインタフェース）作成支援ツールとは異なる。従来のアプリケーションGUI作成支援ツールでは、すでに用意されている部品を配置し、関連づけることによって、GUIを構築していく。そこには、GUIのガイドラインがあり、決められた作法の操作性に基づくように示唆されている。そして生成されるのはソースコードである。本実施例では、C言語等のソースコードを生成せず、UIデザイン検討ツールとして、動作を含めた描画編集とそのシミュレーションを直接おこなう。また、部品は用意されたものを単に配置するのではなく、部品を機能により選択して編集をする。部品の表情と形状は、描画により自由に表現できる。UI画面を作成するデザイナーは、デザインコンセプトにしたがったUI画面の編集とシミュレーションによる動作検討ができる。この場合、UI画面は、アプリケーションと独立して検討を進めることができ、また、アプリケーションに先立つUI画面の検討は、アプリケーションの作法や機能のオーバースペックなどに助言をくわえることが可能である。必要であれば、アプリケーションとの通信メッセージを取り決めることで、アプリケーションプログラマと分業してGUI部を作成することができる。デザイナーは、特別なプログラミング技術をもつ必要がない。図13は本実施例と従来技術の作業プロセスと生成コードの比較のための図であり、（a）は従来のアプリケーションGUI作成支援ツールの場合を示し、（b）は本実施例の場合を示す。

【0048】

【発明の効果】本発明によれば、描画情報だけでなく動

作情報をも設定可能な部品を用いて、動作をも含めたユーザインタフェース部の描画編集を行うことができ、操作に応じて状態の変化する動きのあるユーザインタフェース部の設計を容易に行うことができる。

【0049】また、本発明によれば、作成したユーザインタフェース部の動作シミュレーションを行うことができるので、動作をも含めたデザインの善し悪しを直ちに評価し、デザインの修正を行うという検討作業が可能となり、品質の高いユーザインタフェース部を時間をかけずに設計することができる。更に、本発明はユーザインタフェース部の設計結果を、関係者に対し分かりやすい説得力のあるプレゼンテーションを行うことができる。

【図面の簡単な説明】

【図1】本発明の実施例の構成の概略を示す図

【図2】実施例のハードウェア構成を示す図

【図3】簡単なユーザインタフェースの例を示す図

【図4】アイテムが持つ主な情報を示す図を示す図

【図5】本実施例の6種類のアイテムの例

【図6】編集操作の流れを示す図

【図7】樹木構造によるアイテムの管理を説明する図

【図8】（a）～（c）はフレーム管理による表示の例を示す図

【図9】同じ動作を実現する異なる構造を示すもので、（a）は図3のユーザインタフェースをボタンアイテム2個で構築する場合、（b）はカスケードアイテム1個で構築する場合を示す図

【図10】（a）は利用できる描画図形を示し、（b）は図形プロパティの要素を示す図

【図11】動作シミュレーション部の構成を示す図

【図12】アクションスクリプトの記述子の例を示す図

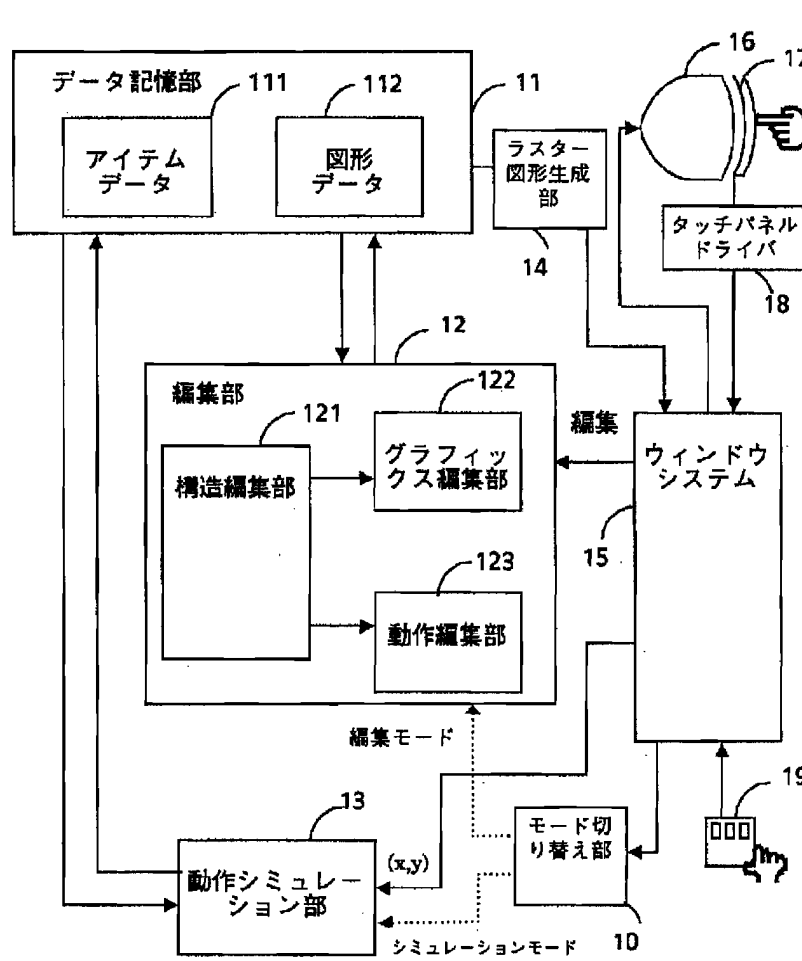
【図13】本実施例と従来技術の作業プロセスと生成コードの比較のための図であり、（a）は従来のアプリケーションGUI作成支援ツールの場合を示し、（b）は本実施例の場合を示す。

【図14】実施例の装置におけるGUIの一例を示す図
【符号の説明】

10…モード切替え部、11…データ記憶部、111…アイテムデータ、112…図形データ、12…編集部、121…構造編集部、122…グラフィックス編集部、123…動作編集部、13…動作シミュレーション部、131…トリガー領域テーブル、132…アイテム管理部、アクションスクリプト解釈部、134…タイマーテーブル、135…変数テーブル、14…ラスタ図形生成部、15…ウィンドウシステム、16…表示部、17…タッチパネル、18…タッチパネルドライバ、19…マウス、1101…通信インタフェース、1102…外部ワークステーション、1103…外部アプリケーション。

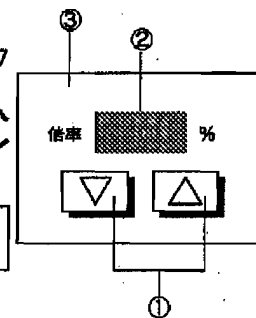
【図1】

【図1】



【図3】

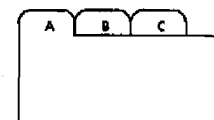
図3. 簡単なUI画面の例



【図5】

図5. 6種類のアイテムの例

①フレームアイテム



②インディケータアイテム

おまちください。

③ボタンアイテム

OFF

④セレクトアイテム

2 3

⑤カスケードアイテム

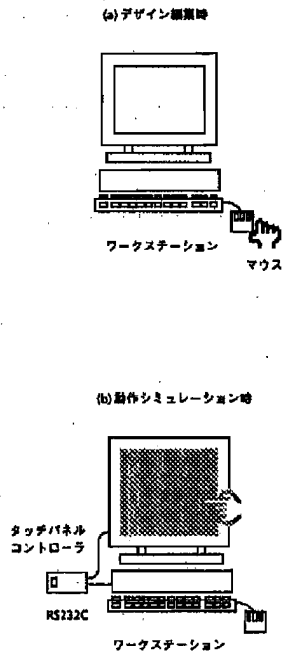


⑥タイマーアイテム

図形の表示はない

【図2】

図2.ハードウェア構成図

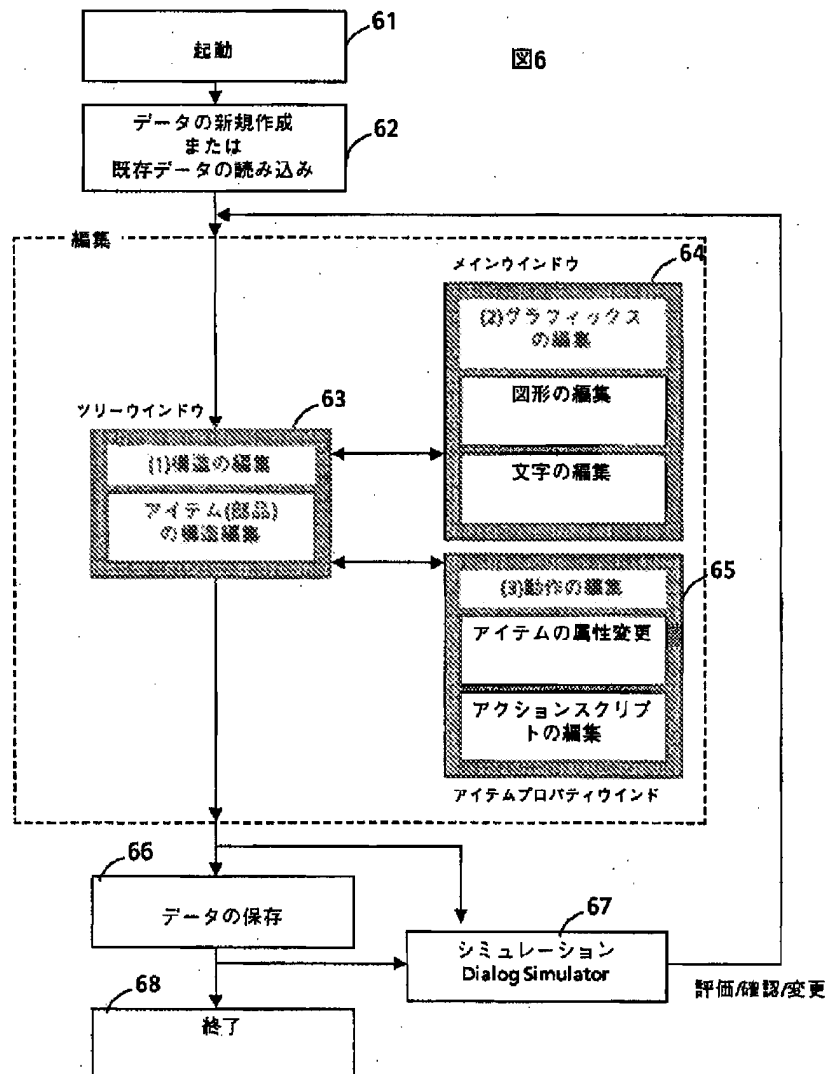


【図4】

図4. 各アイテムが持つ主な情報

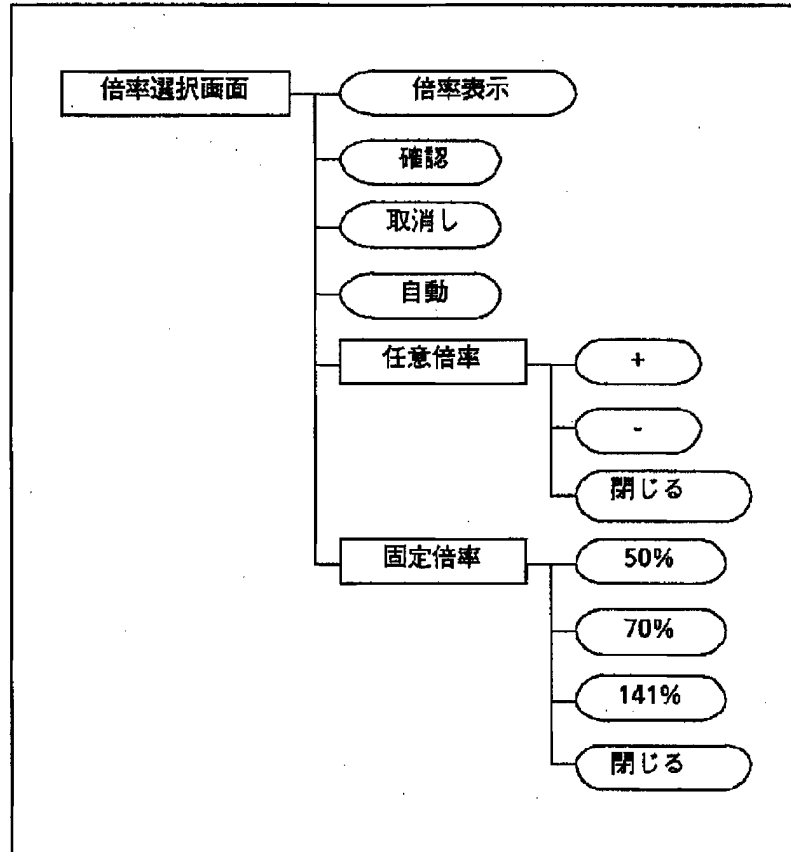
アイテム	情報
フレーム FRAME	図形データ
	下層アイテムへのポインタ
インディケータ INDICATOR	図形データ
ボタン BUTTON	カレント番号(0or1)
	トリガー領域
	2 図形データ
	アクションスクリプト
セレクト SELECTOR	カレント番号(1...N)
	N トリガー領域
	図形データ
	アクションスクリプト
カスケード CASCADE	カレント番号(1...N)
	2 トリガー領域
	サイクリックフラグ
	N 図形データ
	アクションスクリプト
タイマ TIMER	START/STOPフラグ
	呼出し間隔
	アクションスクリプト

【図6】



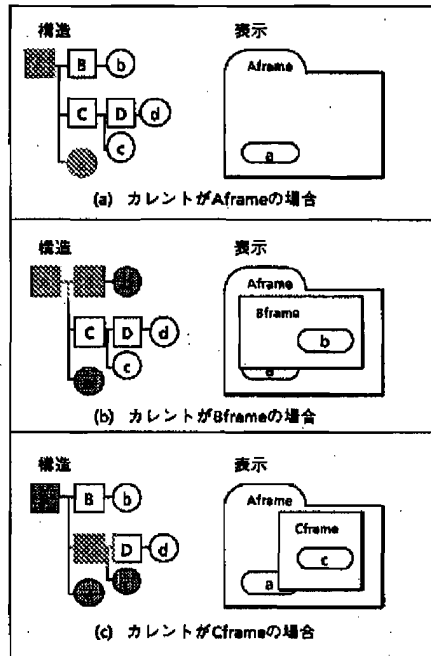
【図7】

図7. 樹木構造によるアイテムの管理



【図8】

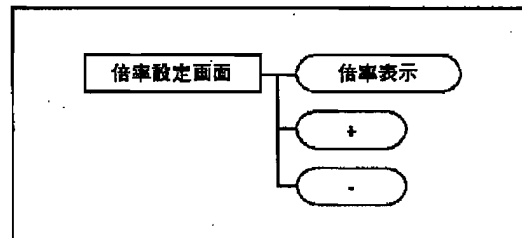
図8. フレーム管理による表示



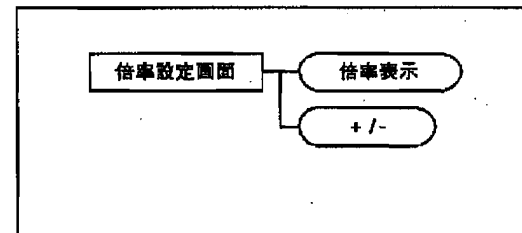
【図9】

図9. 同じ動作を実現する異なる構造

(a) 図3をボタンアイテム2個で構築する場合



(b) 図3をカスケードアイテム1個で構築する場合



【図10】

図10

(a) 利用できる描画図形

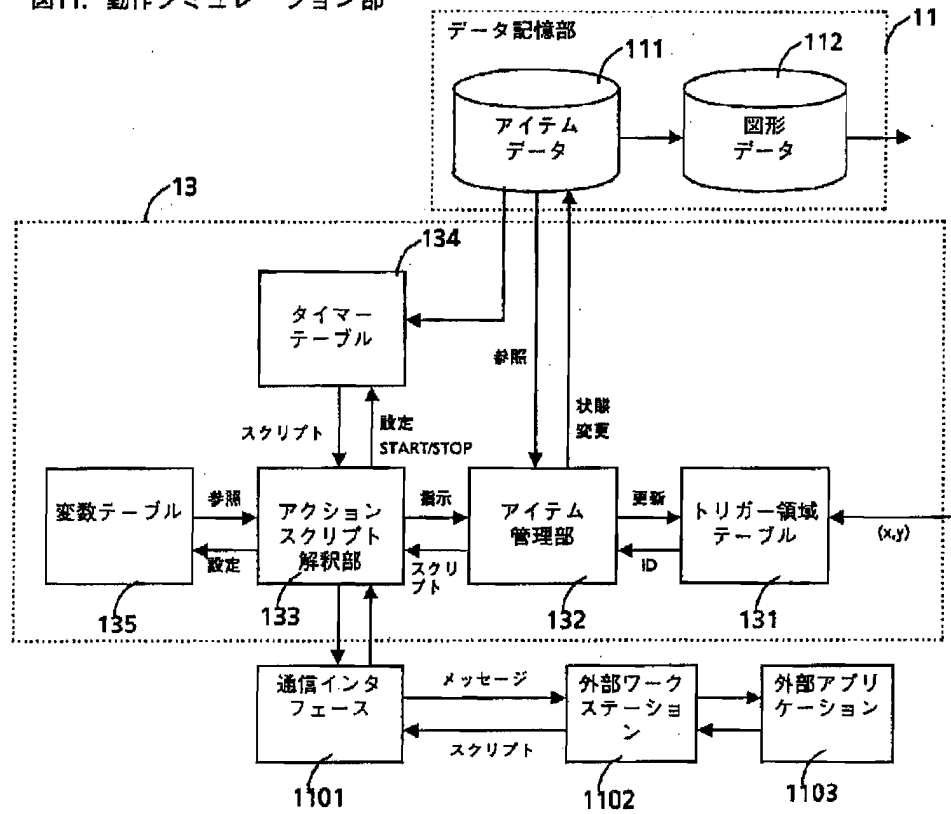
図形	概要
多角線	任意の頂点数の線分
矩形	対角点を指定する四角形
多角形	任意の頂点数の多角形(凹面も可)
円	中心と半径を指定する円
文字	対角点を指定する領域内の文字列(日本語可)
イメージ	対角点を指定する領域内のカラーイメージ

(b) 図形プロパティの要素

要素	概要
線幅	線のドット幅(1以上)
線色	パレットから選択する(256色)
面色	パレットから選択する(256色)
縁装飾	"なし"、"凹"、"凸"、"影"の縁効果
影幅	"影"の縁効果の場合の影のドット幅(1以上)
角R	頂点での面取り半径(ドット数)

【図11】

図11. 動作シミュレーション部



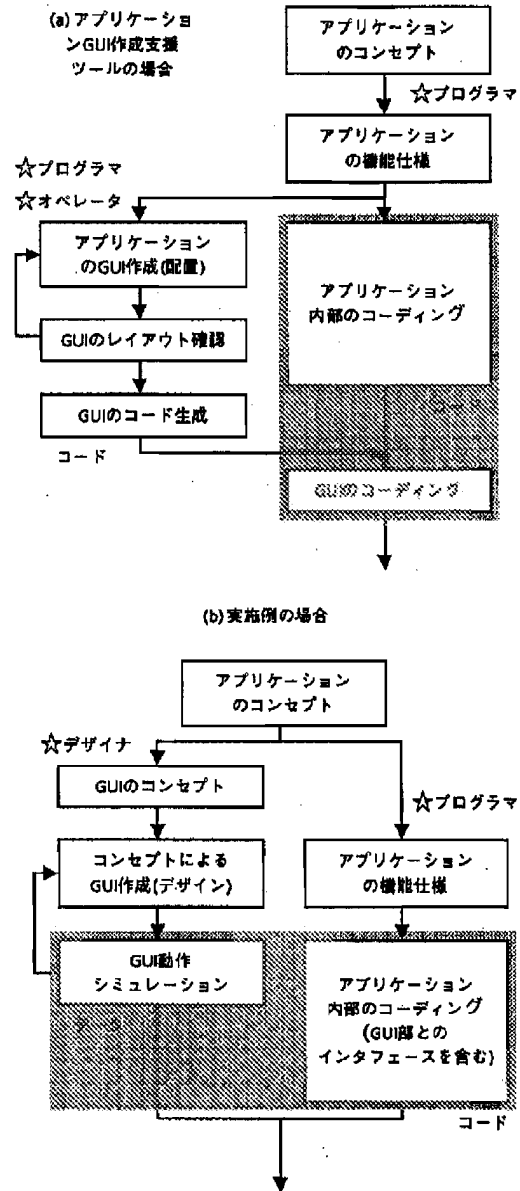
【図12】

図12 アクションスクリプトの記述子(コマンド)一覧

分類	記述子	概要
Frame切替	Jump	カレントFRAMEを移動する
	Popup	子供FRAMEへ移動する(stack-push)
	Close	親FRAMEへ移動する(stack-pop)
図形 ポインタ 切替	Chg	図形のポインタを変更する
	On	図形のポインタを1にするマクロ
	Off	図形のポインタを0にするマクロ
	Default	図形のポインタを初期値にする
変数操作	Set	変数に値を設定する
	Shlrt	変数を10進桁上げる
	Save	変数をローカルからグローバルにする
	Cancel	ローカルスコープの変数を取り消す
間接操作	Do	指定先のスクリプトを評価する
構文	If...else...	解釈の条件分岐
	Return	解釈の中止
入出力	Open	ファイルの指定
	Input	ファイルからの入力
	Output	ファイルへの出力
	Read	ローカルメモリからの入力
	Write	ローカルメモリへの出力
イベント	Flush	表示の強制出力
	Retouch	イベントの再格納

【図13】

図13



[illegible]